# Predictive models based on SVM with structured output

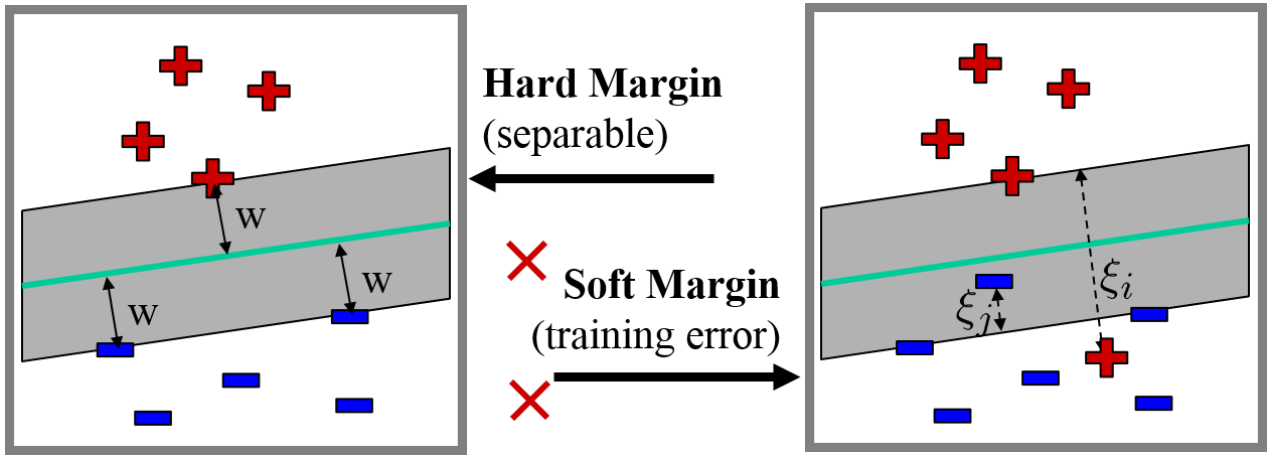
**Jovana Kovačević, Gordana Pavlović-Lažetić**
*Faculty of Mathematics, Belgrade University*
{jovana,gordana}@matf.bg.ac.rs

Matematical Data Science Workshop
Matematical institute, Belgrade, June 21$^{st}$ 2015

## Structural classification vs binarny classification

- binarny classification: $y \in 0, 1$
- structural classification: $y \in Y$, $Y$ is structured (sequence, tree, graph, . . . )
- SSVM - generalization of SVM method on structural output (*T. Joachims et al, 2004*)

## SVM




$f(\boldsymbol{x}) = \langle \boldsymbol{x}, \boldsymbol{w} \rangle + b$

$y = sgn(f(\boldsymbol{x}))$

Minimize

$$\frac{||\boldsymbol{w}||^2}{2} + \sum_{i=1}^{n} \xi_i$$

such that $y_i \cdot f(\boldsymbol{x_i}) \geq 1 - \xi_i$ for each training example $(\boldsymbol{x_i}, y_i)$

## Generalization of SVM on structural output: SSVM

- Goal: for training set $(\boldsymbol{x_1}, \boldsymbol{y_1}), \ldots, (\boldsymbol{x_n}, \boldsymbol{y_n})$, find a function $f : X \to Y$

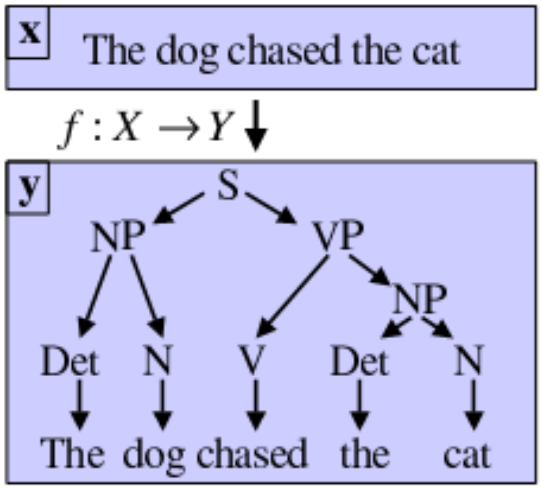
- Idea: $F : X \times Y \to \mathbb{R}$

$$f(\boldsymbol{x}) = \operatorname{argmax}_{y \in Y}(F(\boldsymbol{x}, \boldsymbol{y}))$$

## SSVM

- let us assume: $F(\boldsymbol{x}, \boldsymbol{y}) = \langle \Psi(\boldsymbol{x}, \boldsymbol{y}), \boldsymbol{w} \rangle$

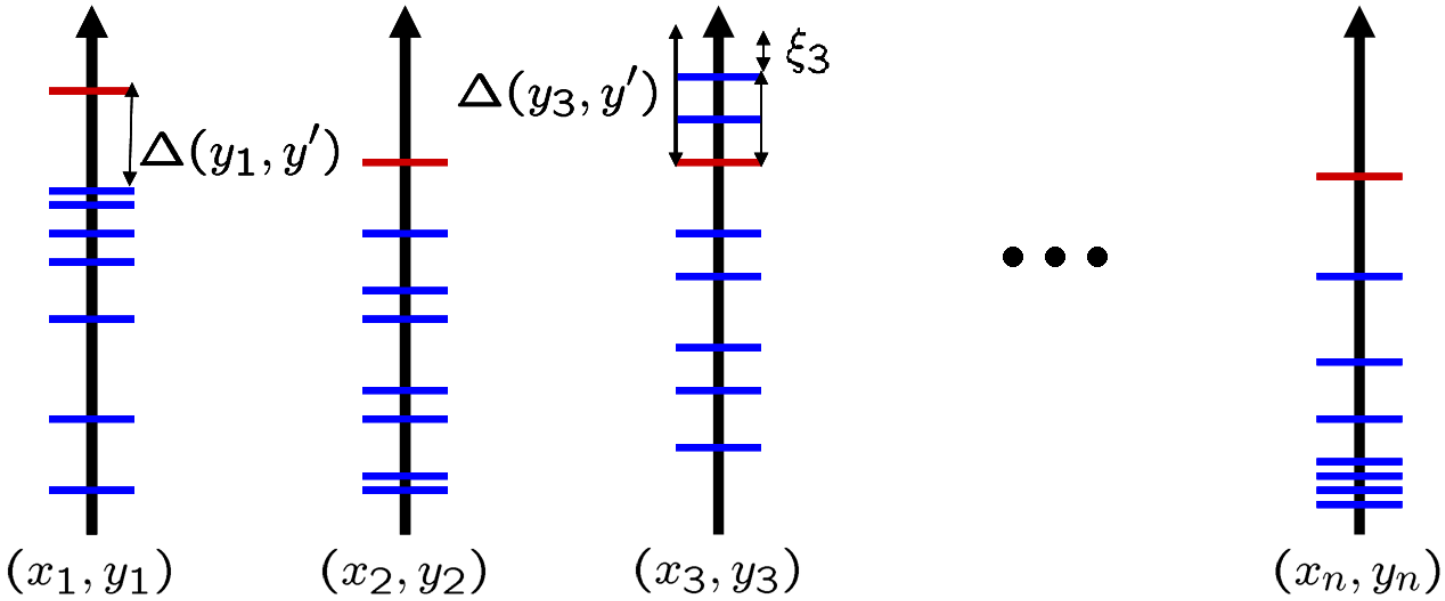
- function $\Psi$ is joint representation of input data $\boldsymbol{x}$ and output data $\boldsymbol{y}$




$$\Psi(\mathbf{x}, \mathbf{y}) = \begin{pmatrix} 1 & S \to NP\,VP \\ 0 & S \to NP \\ 2 & NP \to Det\ N \\ 1 & VP \to V\ NP \\ \vdots & \\ 0 & Det \to dog \\ 2 & Det \to the \\ 1 & N \to dog \\ 1 & V \to chased \\ 1 & N \to cat \end{pmatrix}$$

## SSVM - Margin in structured setting

- We would like coefficients $w$ to be such that $\langle \Psi(\boldsymbol{x}, \boldsymbol{y}), \boldsymbol{w} \rangle$ is highest for correct $\boldsymbol{y}$



## SSVM formulation of optimization problem

$$\min_{\boldsymbol{w},\boldsymbol{\xi}} \frac{1}{2}||\boldsymbol{w}||^2 + \frac{C}{n}\sum_{i=1}^{n}\xi_i$$

subject to $\forall i \in \{1,\ldots,n\}$

$$\forall \boldsymbol{y} \in Y : \langle \boldsymbol{w}, \Psi(\boldsymbol{x_1}, \boldsymbol{y_1})\rangle - \langle \boldsymbol{w}, \Psi(\boldsymbol{x_1}, \boldsymbol{y})\rangle \geq \Delta(\boldsymbol{y_1}, \boldsymbol{y}) - \xi_1$$

$$\ldots$$

$$\forall \boldsymbol{y} \in Y : \langle \boldsymbol{w}, \Psi(\boldsymbol{x_n}, \boldsymbol{y_n})\rangle - \langle \boldsymbol{w}, \Psi(\boldsymbol{x_n}, \boldsymbol{y})\rangle \geq \Delta(\boldsymbol{y_n}, \boldsymbol{y}) - \xi_n$$

- quadratic optimization with linear constraints
- $n|Y|$ constraints!

**Algorithm 1** Struct SVM learning algorithm (Cutting plane)

1: $W_i = \emptyset$
2: **repeat**
3: **for** $i = 1, \ldots n$ **do**
4: out of all $\hat{\boldsymbol{y}}_{\boldsymbol{i}} \in Y$, find $\hat{\boldsymbol{y}}_{\boldsymbol{i}}$ that violates constraint $i$ most (argmax problem: $\arg\max_{\hat{\boldsymbol{y}}_{\boldsymbol{i}} \in Y}(\Psi(\boldsymbol{x_i}, \hat{\boldsymbol{y}}_{\boldsymbol{i}}) + \Delta(\boldsymbol{y_i}, \hat{\boldsymbol{y}}_{\boldsymbol{i}}))$)
5: **if** $\hat{\boldsymbol{y}}_{\boldsymbol{i}}$ violates constraint for more than $\epsilon$ **then**
6: update set $W_i$: $W_i = W_i \cup \hat{y}$
7: solve optimization problem with constraints:

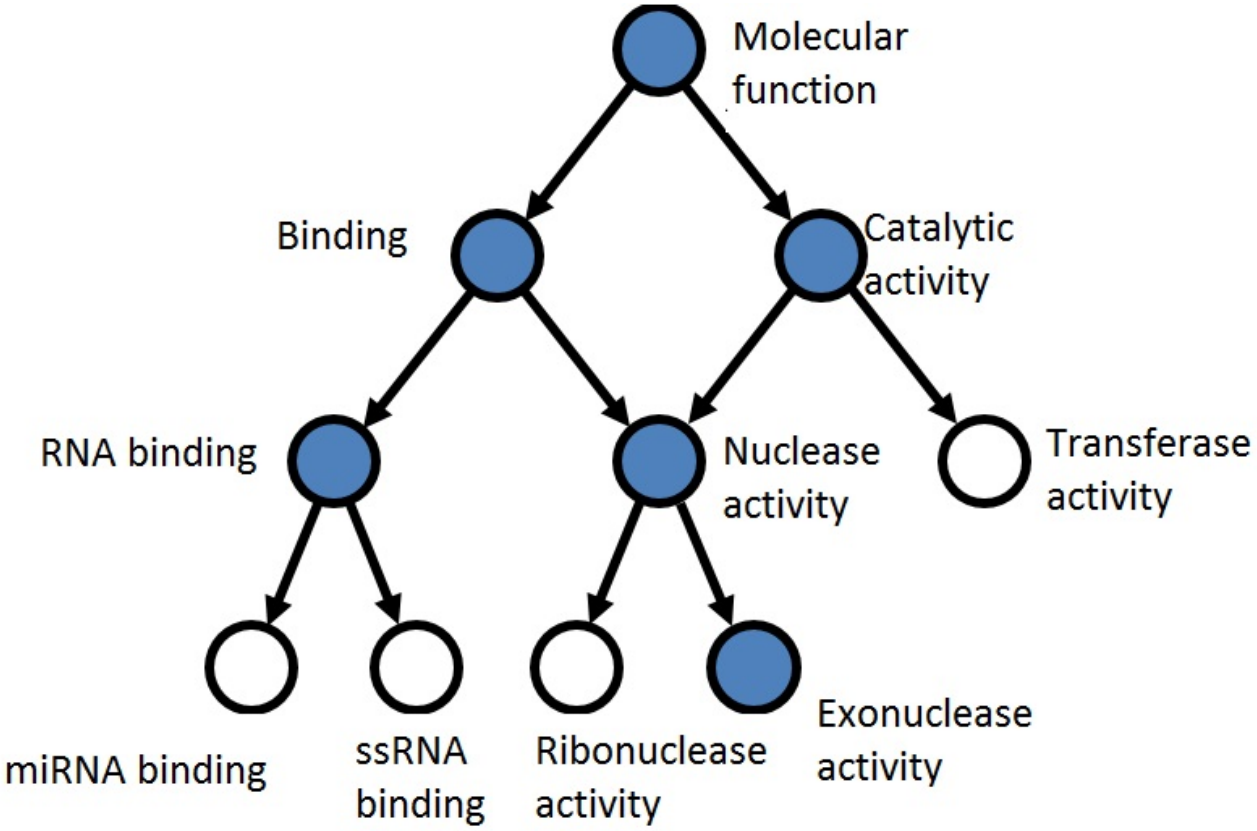
$$\forall \boldsymbol{y} \in W_1 : \langle \boldsymbol{w}, \Psi(\boldsymbol{x_1}, \boldsymbol{y_1}) \rangle - \langle \boldsymbol{w}, \Psi(\boldsymbol{x_1}, \boldsymbol{y}) \rangle \geq \Delta(\boldsymbol{y_1}, \boldsymbol{y}) - \xi_1$$

$$\ldots$$

$$\forall \boldsymbol{y} \in W_n : \langle \boldsymbol{w}, \Psi(\boldsymbol{x_n}, \boldsymbol{y_n}) \rangle - \langle \boldsymbol{w}, \Psi(\boldsymbol{x_n}, \boldsymbol{y}) \rangle \geq \Delta(\boldsymbol{y_n}, \boldsymbol{y}) - \xi_n$$

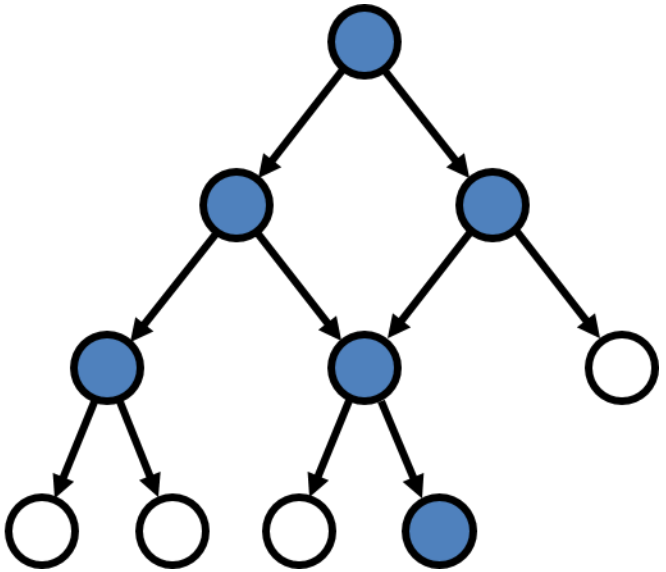
8: **until** each set $W_i$ is unchanged

# Protein function prediction



# Protein function prediction

**>sp|P04637|P53_HUMAN**
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIE
QWFTEDPGPDEAPRMPEAAPPVAPAPAAPTPAAPAPAPSWPLSSSVPSQKT
YQGSYGFRLGFLHSGTAKSVTCTYSPALNKMFCQLAKTCPVQLWVDSTPPP
GTRVRAMAIYKQSQHMTEVVRRCPHHERCSDSDGLAPPQHLIRVEGNLRVE
YLDDRNTFRHSVVVPYEPPEVGSDCTTIHYNYMCNSSCMGGMNRRPILTII
TLEDSSGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGEPHHELPPGSTKR
ALPNNTSSSPQPKKKPLDGEYFTLQIRGRERFEMFRELNEALELKDAQAGK
EPGGSRAHSSHLKSKKGQSTSRHKKLMFKTEGPDSD

## Implementation of $\Psi$ function

- input: histogram of tetragrams
- output: sparse 0/1 vector, $y_i(j) = 1$ if protein $i$ contains node $j$ in its function DAG
- dimension of vector $\Psi$: total number of tetragrams $\times$ total number of nodes that appear in dataset

$$\Psi(\boldsymbol{x}, \boldsymbol{y}) = [\underbrace{\mathbf{0}}_{y_1}, \ldots, \underbrace{\mathbf{0}}_{y_{a-1}}, \underbrace{\boldsymbol{x}}_{y_a}, \underbrace{\mathbf{0}}_{y_{a+1}} \ldots, \underbrace{\mathbf{0}}_{y_{b-1}}, \underbrace{\boldsymbol{x}}_{y_b}, \underbrace{\mathbf{0}}_{y_{b+1}}, \ldots, \underbrace{\mathbf{0}}_{y_{c-1}}, \underbrace{\boldsymbol{x}}_{y_c}, \underbrace{\mathbf{0}}_{y_{c+1}}, \ldots, \underbrace{\mathbf{0}}_{y_k}]$$

**Algorithm 2** Augmented inference algorithm

**Input**: training instance $(\boldsymbol{x_i}, \boldsymbol{y_i})$
**Output**: $\boldsymbol{y_{best}}$ that maximizes $H(\boldsymbol{x_i}, \boldsymbol{y})$ over $\boldsymbol{y} \in Y$
**Initialization**: $L = \{\boldsymbol{y_{root}}\}, \boldsymbol{y_{best}} = \emptyset, H_{best} = -\infty$
**repeat**
    $\boldsymbol{y_{head}}$ := first element from $L$
    $Y_{ext}$ := all extensions of $\boldsymbol{y_{head}}$ by one node
    **for** each $\boldsymbol{y_{ext}} \in Y_{ext}$ **do**
        **if** $i(\boldsymbol{y_{ext}}) \geq imax$ **then**
            **continue**
        insert $\boldsymbol{y_{ext}}$ in sorted linked list $L$
        **if** $H(\boldsymbol{y_{ext}}) > H_{best}$ **then**
            update $\boldsymbol{y_{best}}, H_{best}$
    remove $\boldsymbol{y_{head}}$ from $L$
    increment $step$
**until** $step > smax$ or $L$ is empty

## Parameter imax

- model ontology with Bayesian network:

$$Pr(T) = \prod_{v \in T} Pr(v|\mathcal{P}(v))$$

- information content of a DAG $T$

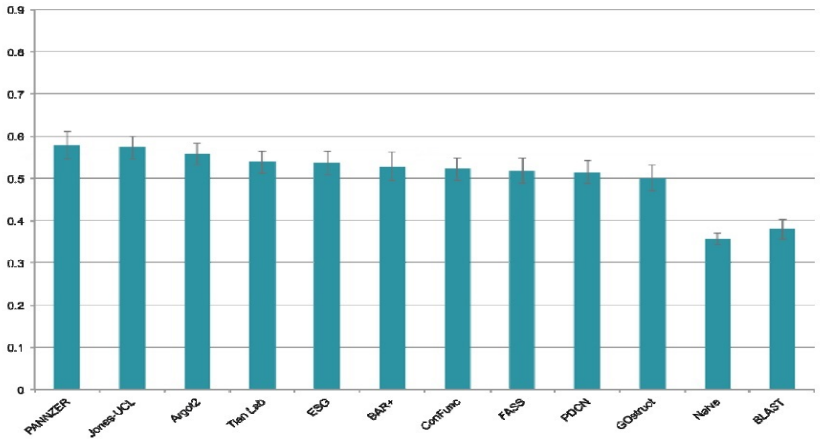
$$i(T) = \log \frac{1}{Pr(T)}$$
$$= \sum_{v \in T} \log \frac{1}{Pr(v|\mathcal{P}(v))} = \sum_{v \in T} ia(v)$$

- information assertion of a node $v$ - rare nodes have high $ia(v)$

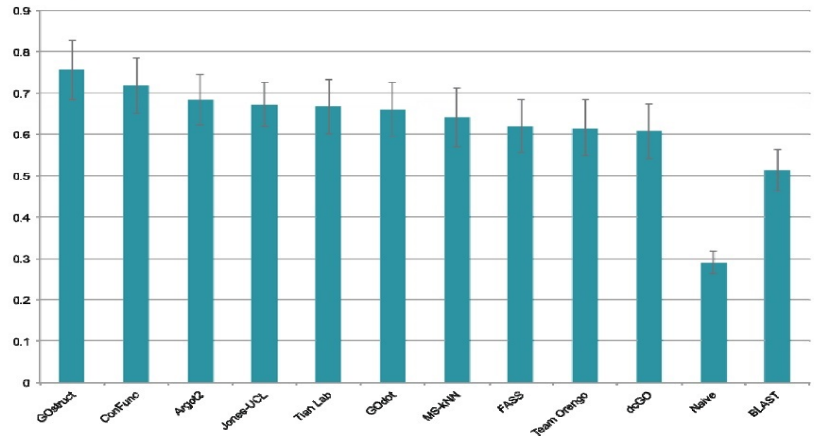
*Radivojac,Clark, 2013*

# Results - comparison with CAFA

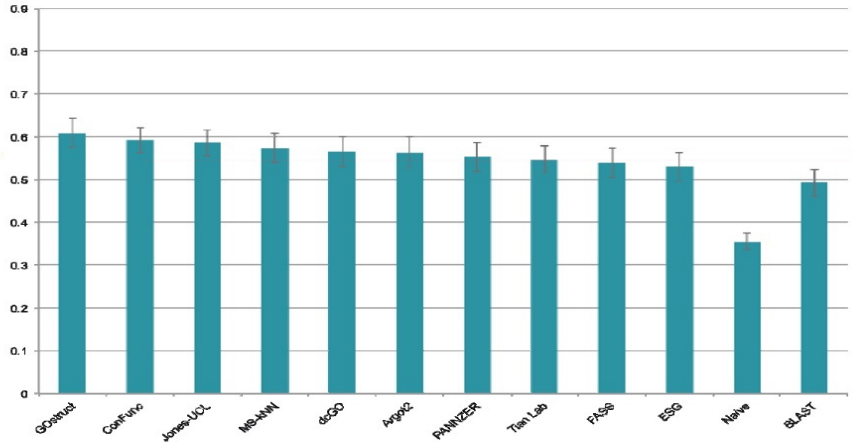
Supplementary Figure 7C: Molecular Function – *H. sapiens*.



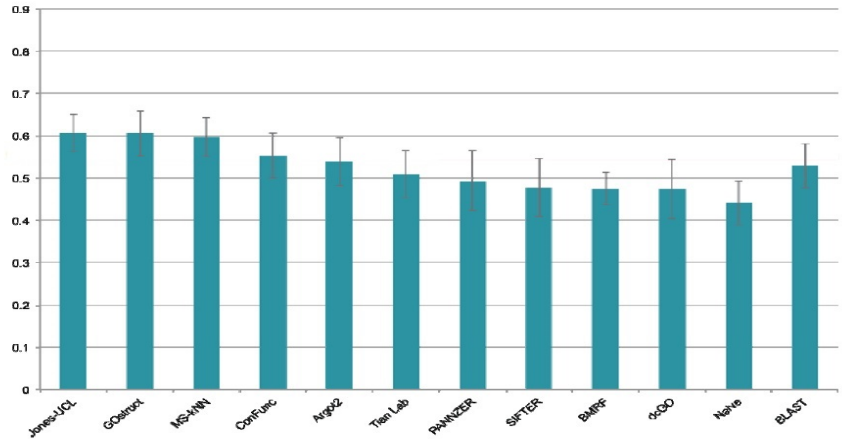

Supplementary Figure 7A: Molecular Function – *A. thaliana*.




Supplementary Figure 7D: Molecular Function – *M. musculus*.

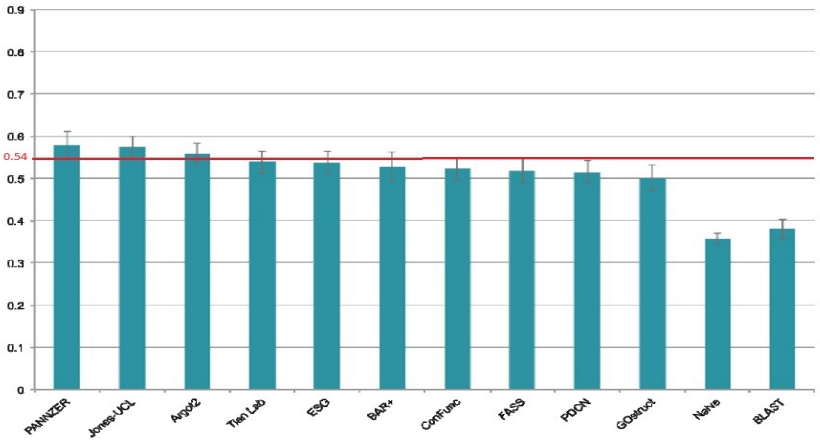



Supplementary Figure 7E: Molecular Function – *R. norvegicus*.

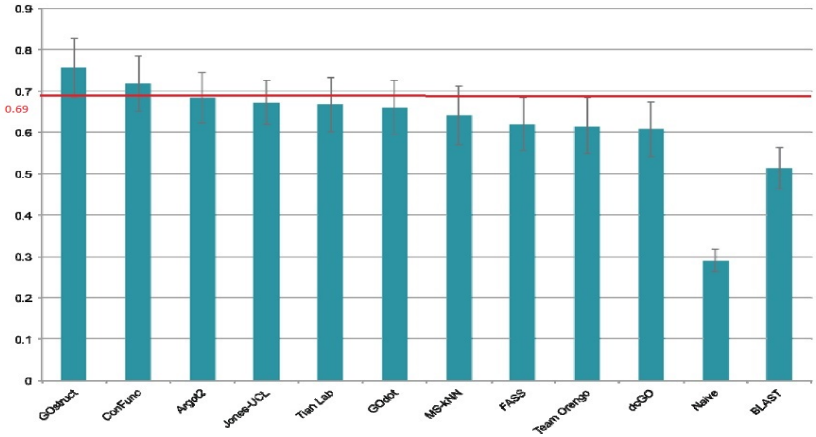


# Results - comparison with CAFA

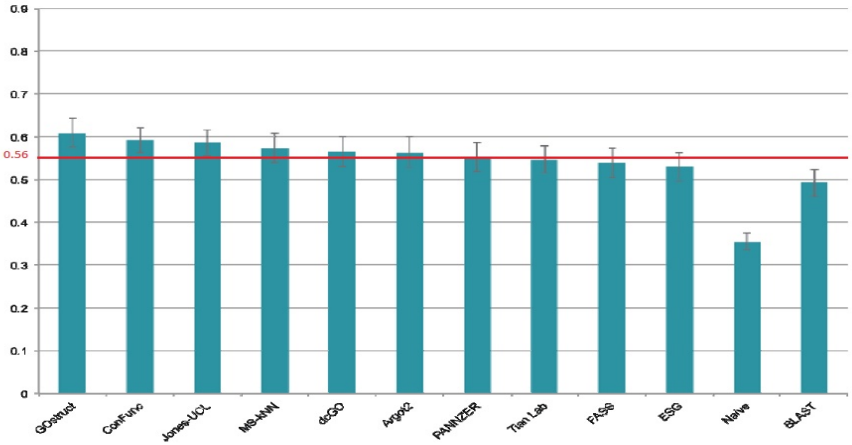
Supplementary Figure 7C: Molecular Function – *H. sapiens*.



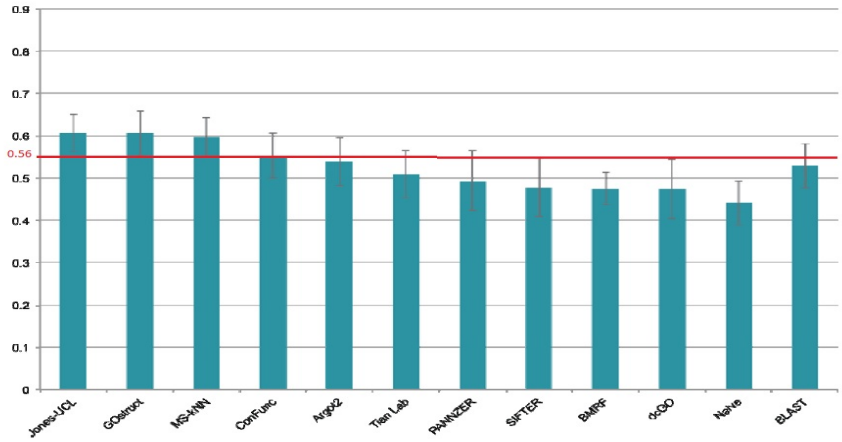

Supplementary Figure 7A: Molecular Function – *A. thaliana*.




Supplementary Figure 7D: Molecular Function – *M. musculus*.

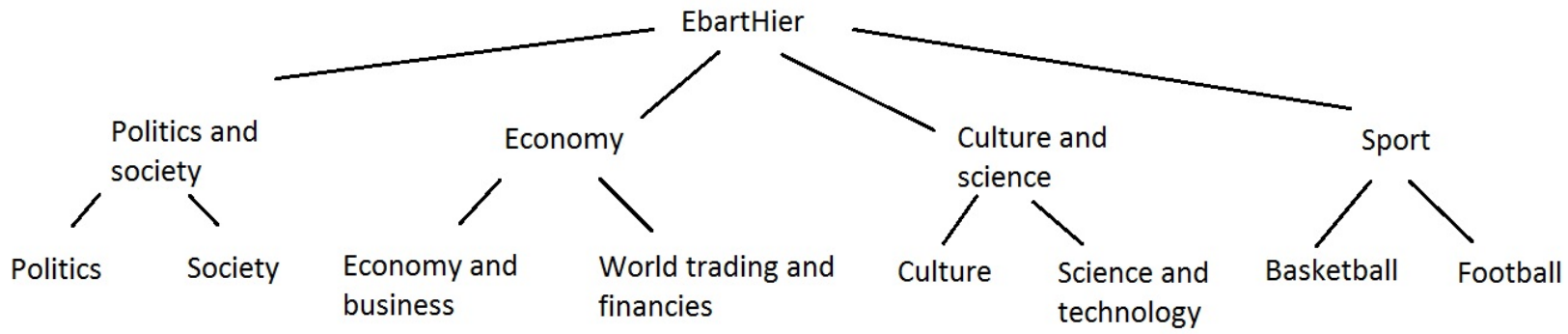



Supplementary Figure 7E: Molecular Function – *R. norvegicus*.



## Text classification

- hierarchical classification and multiclassification of text corpora in different languages



## Text classification

- each text is represented by ngrams of bytes, $n \in \{2, 3, 4, \ldots\}$
- input: each position of vector $x$ corresponds to its tf-idf statistics in the training set
- output in multiclassification: a class from the set $\{1, 2, \ldots, N\}$, where $N$ is total number of classes
- output in hierarchical classification: a path from the root to the corresponding leaf

## Results

| | size of dataset | number of classes | language | our method | result | best known result |
|---|---|---|---|---|---|---|
| Ebart | 60637 | 8 | Serbian | hier<br>flat | 90.17<br>**90.43** | Graovac, 2012, knn: 88.5 |
| Tancorp | 14150 | 60 | Chinese | hier<br>flat | **87.38**<br>87.19 | Li, 2010, knn: 79.37 |
| mesleh-10 | 7842 | 20 | English | flat | **93.44** | Mesleh, 2011, SVM: 91.41 |
| 20 newsgroups | 20000 | 8 | Arabic | flat | **83.75** | Lan et al, 2009, SVM: 80.81 |

Thank you for your attention! Questions?